Certification Paths with the Java™ 2 Platforms

A Presentation to the Federal PKI Technical Working Group

September 14, 2000

SML-2000-0492


Sun
microsystems

Certification Path Processing with the Java™ 2 Platforms

Steve Hanna
Senior Staff Engineer
Sun Microsystems, Inc.

Presentation Outline

- Background
- CertPath API and RI Overview
- CertPath API and RI Details
- Conclusions

Background

Java:

Programming Language

AND

Platforms

Java 2 Platforms

- Java 2 Platform, Micro Edition (J2ME™)
- Java 2 Platform, Standard Edition (J2SE™)
- Java 2 Platform, Enterprise Edition (J2EE™)

Java Community Process℠

- Process used for all significant changes to the Java 2 Platforms
- Steps
  - Initiation
  - Community Draft
  - Public Draft
  - Maintenance
- Guided by multi-party Executive Committee

J2SE Certificate Support

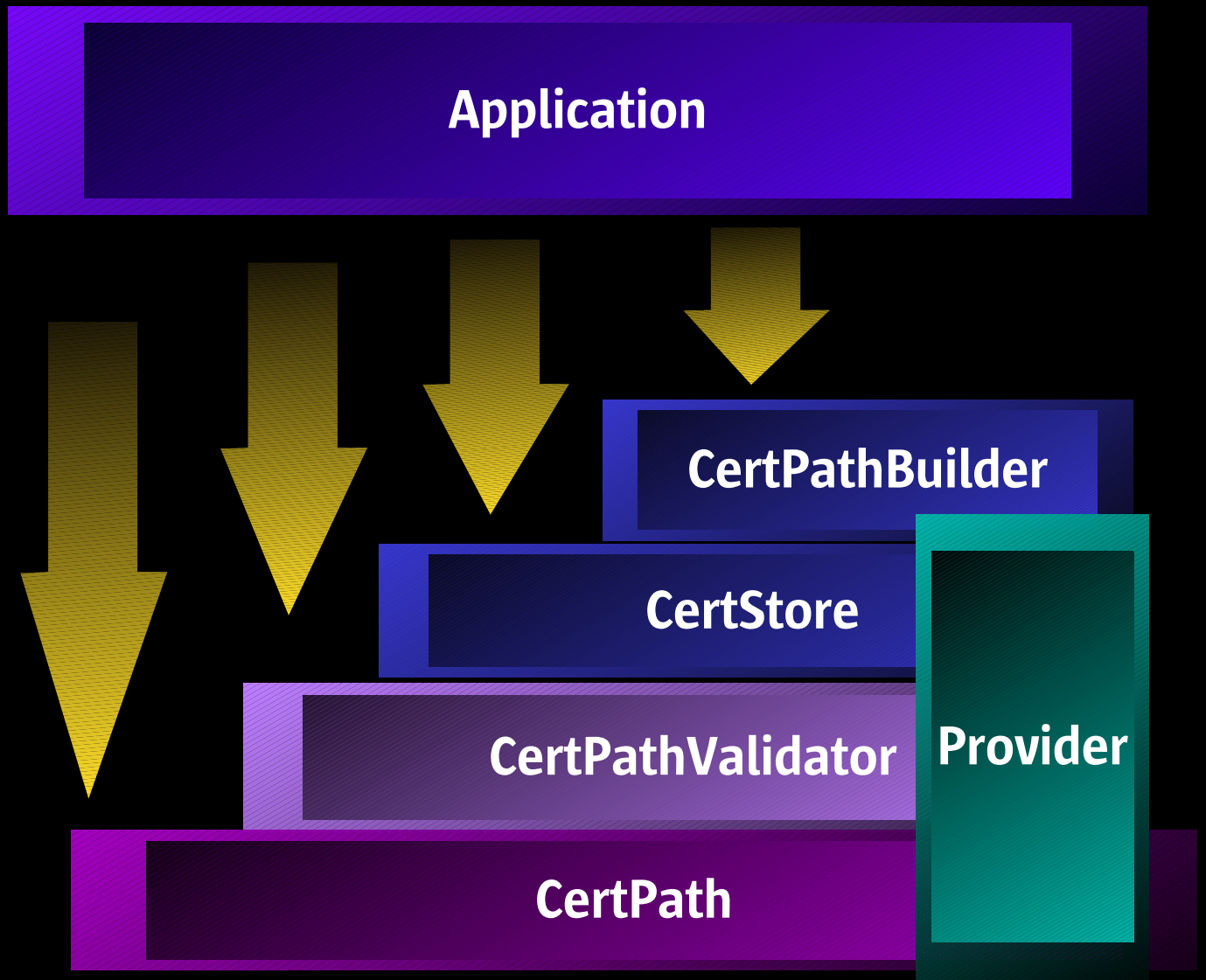
Past and Present

- JDK™ 1.1 (1997)
  - Minimal certificate support, primarily for signed code
- JDK 1.2 (1998)
  - APIs for parsing X.509 certificates
  - Tools for managing certificates and private keys
- JDK 1.3 (2000)
  - Minor changes

New CertPath APIs and RI

- Provide PKIX-compliant certification path validation and development
- Initially developed at Sun Labs
- API now in Java Community Process (JSR 55)
- API and RI will probably be included in JDK 1.4 (Merlin, 2001)
- RI = Reference Implementation

CertPath Architecture



Provider-based Architecture

- Applications use standard API
- Can have multiple implementations installed

CertPath API and RI Details

CertPath

- Ordered List of Certificate objects
- RI supports X.509 certificates
- CertificateFactory enhanced to support creating CertPaths
- RI can read/write PKCS#7 format

CertPathValidator

- result = cpv.validate(path, params);
- Relying Party (RP) supplies CertPath and parameters
- Validator attempts to validate path
- On success, returns result
- On failure, throws exception with details

PKIX Validation

- Parameters include:
  - Trust anchors, initial policies, as-of date, CertStores (for retrieving CRLs), constraints on target certificate (subject alt name, etc.), other miscellanous items
- Results include:
  - Policies, policy qualifiers

CertPathBuilder

- result = cpb.build(params);
- RP supplies parameters
- Builder attempts to build validated path
- On success, returns result
- On failure, throws exception

PKIX Building

- Parameters include:
  - Validation parameters (trust anchors, etc.)
  - Maximum path length
- Results include:
  - Validated CertPath
  - Policies, policy qualifiers

Optimizations in RI

- Validate path as you build it
- Eliminate certs based on validation
- Observed:
  - Building reverse allows for more efficient certificate elimination than building forward
    - Name Constraints
    - Policy Processing
    - Signature Processing
  - Loops are not always bad (policy mapping)
  - Paper submitted to NDSS

CertStore

- certs = cs.getCertificates(selector);
- RP supplies selector
- Store returns selected certificates
- RI supports LDAP and cache

Extensibility

- Support for private extensions
- Support for custom validation checks
- Support for different revocation checking
- Support for different providers
- Support for various certificate and CRL repositories

More Extensibility

- Support for non-X.509 certificates
- Support for non-PKIX validation
- Support for changes to PKIX validation

Testing

- Tested with our own automated test suite
- Plan to test with BCA directory soon
- Will work on testing matrix to take RFC 2459 to Draft Standard

Benefits

- Solid certification path processing soon available to all Java™ technology-based code
- Will eventually be used by other J2SE functions (JSSE & code signing)
- Supports many trust models (bridge, mesh, hierarchy, etc.)
- Checks revocation status
- Useful for 2459 interoperability testing

Lessons Learned

- Weighting useful in dense meshes, but not dependable
- Constraints (especially name constraints) help with this
- Advantages of reverse building in dense meshes
- Involve many experienced parties ASAP

To Be Done

- Interoperability Testing
- Complete JCP
- Ship with JDK 1.4
- Use for JSSE and code signing
- Integrate with Java™ technology-based applications!
- Continue evangelizing PKIX-compliant validation and building
- Continue research into building

STEVE HANNA

steve.hanna@sun.com

We're the
dot in .com™


We’re the
dot in .com™
Sun
microsystems